REMARKS

Claims 1, 3-39 and 41-64 are currently active.

Claims 48-64 have been added. Antecedent support for Claims 48-64 is found in Claims 1, 4, 5, 6 and 7.

Antecedent support for the amendments to Claim 1 and Claim 14, and new Claims 42-47 is found on page 12, lines 9 and 10, 17-25 and page 13, lines 1-3, 12-16 and 24.

The claims have been amended to obviate the objections to the claims.

The Examiner has rejected Claims 1, 3-35 and 38-41 as being unpatentable over Fridella incorporating Vahalia in view of Carns. Applicants respectfully traverse this rejection in view of the amendments to the claims and that Carns specifically teaches not to use PVFS with NFS.

Referring to Fridella, there is disclosed management of file-modification time attributes in a processor fileserver system. Fridella teaches a network file server includes multiple data mover computers 115, 116 and 117, each of which manages a respective

filesystem. Each data mover computer also functions as a fileserver for servicing client requests for access to the filesystems. Each data mover computer has a respective port to a data network 111 having a number of clients including workstations. The network file server includes a cached disc array 114. The clustering of the data movers as a front end to the cached disk array provides parallelism and scalability. The data movers may communicate with each other over a dedicated dual-redundant ethernet connection 118. See paragraphs 26 and 27.

In the network file server, the locking information for each filesystem is managed exclusively by only one of the data movers. This exclusive relationship is referred to by saying each filesystem has a respective data mover that is the owner of the filesystem. The owner of a filesystem is said to be primary with respect to the filesystem, and other data movers are said to be secondary with respect to the filesystem. See paragraph 28.

Each client may access any of the file systems through any one of the data mover computers but if the data mover computer servicing the client does not own the filesystem to be accessed, then a lock on at least a portion of the filesystem to be accessed must be obtained from the data mover computer that owns the filesystem to be accessed. See paragraph 29. The teaching of Fridella not only is silent regarding "the servers utilizing a striped file system," as found in Claim 1, but it teaches away from this limitation because

Fridella requires every secondary data mover to obtain a lock from the primary data mover on at least a portion of the file system to be accessed by the secondary data mover. This requirement of the teaching of a secondary data mover which must obtain a lock from the primary data mover every time the secondary data mover wants to access the file system makes it completely unfeasible to have "servers utilizing a striped file system for storing data." It places a material step (obtain a lock) that would interfere with if not totally disrupt an operation on a different portion of the data as a single strip in every server because an operation regarding the data operates on the single strip of each server, as found in amended Claim 1. In fact, applicants respectfully submit a striped file system simply cannot be used with the lock architecture taught by Fridella. No skilled artesian would even consider such an architecture for striping.

Fridella teaches when a write operation will change the meta-data of a file, the meta-data must be managed in a consistent fashion, in order to avoid conflict between the data mover owning the file, and the data mover performing the write operation. When a secondary data mover performs a write operation that changes the meta-data of a file, the new meta-data is written to the primary data mover. See paragraph 30. This teaching again teaches away from "the servers utilizing a striped file system" because every strip written to a secondary server would require new meta-data written to the primary data mover, which would make

striping totally impractical, for the same reasons explained above. It would totally disrupt operating on the single strip of each server, as found in Claim 1, if not make it impossible.

Fridella teaches it is possible for the secondary data movers to update the file-modification time attribute concerning a file in a consistent fashion without always accessing the primary data mover clock. The clocks of the primary and secondary data mover need not be synchronized. The secondary clocks cannot simply be used to set the file-modification time attributes, because the clock skew between the multiple secondary data movers writing to the same file would violate the second consistency requirement. On the other hand, the primary clock cannot simply be used unless the file-modification time is updated for each asynchronous write. It is possible though for a secondary data mover to update the file-modification time attribute in a consistent fashion using a hybrid method that comprises the file-modification time attribute based on the clock of the primary data mover and a timer of the secondary data mover. See paragraph 36.

When a server performs an asynchronous write for a client, the server returns an updated file as a modification time attribute. If the server is the primary server, the updated-file-modification time can simply be the time of its local clock. If the server is the secondary server, then the updated filed modification time is the sum of the local timer and a local value of the primary clock having been stored in local memory of the secondary server.

When the secondary server performs a second asynchronous write to the filesystem, it computes an updated filed-modification time by adding the stored clock time and the present value of the timer, and returns the file-modification time to the client. See paragraphs 38 and 39. This is one of the focuses of Fridella, which has nothing to do with a striped file system.

It is clear from the above description, there is no teaching or suggestion whatsoever of "the servers utilizing a striped filesystem for storing data," as found in Claim 1 of applicants.

To deal with the fact that Fridella fails to teach anything all about striping, the Examiner cites Vahalia through incorporation by reference. It is respectfully submitted that Fridella cites Vahalia in column 1, under the background of the invention section. Fridella specifically cites Vahalia as explaining a solution for data consistency problems if concurrent client access to a file is permitted to more than one data mover. Fridella cites Vahalia for the solution of locking information can be stored in the cached disk array, or are cached in the data mover computers if a cached coherency scheme is used to maintain consistent locking data in the caches in the data mover computers. See paragraph 5 of Fridella. First, in this context, there is no teaching or explanation of how the teachings of Vahalia would be used in the description of the system taught in Vahalia beginning on paragraph 26, let alone how striping would be used with Vahalia. To reiterate, the specific teaching by Fridella incorporating

Vahalia has nothing at all to do with the modification or use of the actual system taught by Vahalia; it is simply to provide an example of a solution for a specific problem identified in the background of the invention by Fridella.

Secondly, and even more importantly, a review of figures 2, 3 and 5, and the supporting text for these figures, as well as the specific textual citations by the Examiner, such as column 7, lines 61-66, which the Examiner relies upon for Vahalia teaching striping, in fact fails to teach anything at all about striping. It is respectfully submitted, that the Examiner is reading the limitations of applicants' Claim 1 into the teachings of Vahalia when they are simply not there. Applicants are unsure if the Examiner is interpreting the architecture of Vahalia in regard to the sections that he cites, as capable of performing striping, and thus meets the limitations of applicants' claimed invention, but this is contrary to patent law.

More specifically, a review of figures 2 and 3 simply show the general architecture of the network file server 20 and a integrated cached disk array 23, respectively. Beginning at column 5, lines 30 of a Vahalia, figure 2 is discussed. Vahalia teaches in the preferred mode of operation, to archive data from a file from the network to tape, one of the data movers receives the file from the network and pre-stages the file to the cached disk array at a high rate limited by the network transmission rate. Then one of the data movers destages the file from the cached disk array to an associated one of the read/write stations at a tape

device speed. For most applications, pre-staging to disk can be done immediately, and staging from disk to tape including sorting of files on to respective tape disk can be done as a background operation or at night, when the load on the network file server is at a minimum. In this fashion, the cached disk array can absorb a high dataflow aggregation from tens or hundreds of network links streaming from multiple sites, and balances this load on the read/write stations. See column 5, lines 48-63. From this description of figure 2 in its operation, it is clear that Vahalia only teaches sorting the files onto respective tape cassettes, which has nothing at all to do with striping.

In regard to figure 3, Vahalia teaches that the channel directors access data in the cached memory in response to a request from its associated data mover. If data to be read by a channel director is not found in a cached memory, one of the disk directors and disk adapters transfers or stages the data from the disk array to the cached memory. In a background process, the disk directors and disk adapters also write-back data from the cached memory to the disk array, after the channel directors write data to the cached memory. In addition to providing intermediate storage for the data transferred between the channel directors and the disk directors, the cached memory also provides intermediate storage for control information transferred among the channel directors and disk directors.. See column 6, lines 20-33. Again, it is respectfully submitted that figure 3 and the associated text has nothing at all to do with striping.

On page 4 of the Office Action, at the beginning of the second paragraph, the Examiner states that the servers utilizing a striped filesystem for storing data (Vahalia's figure 5, column 7 lines 61-66) further discloses data blocks which are striped across disks and disk array are being tracked by physical filesystems and each server. A review of figure 5, and column 7, lines 61-66 of Vahalia, states figure 6 is a specific example of software modules of figure 5. A conventional UNIX filesystem is a physical filesystem exported on to the network using NFS. The filesystem switch that directs client NFS requests to the intended physical filesystem is implemented using a standard virtual file-system interface. Again, it is respectfully submitted the specific teachings of column 7, lines 61-66 have nothing at all to do with striping.

Accordingly, simply speaking, Vahalia adds nothing to the teachings of Fridella in regard to the limitations in Claim 1 of applicants regarding striping.

The Examiner cites Carns for disclosing a virtual filesystem in which data blocks of the filesystem can be striped across the I/O node/disks, see figure 1, section 3.1 second and third paragraphs of Carns. Referring to this citation in Carns, Carns teaches that the PVFS files are striped across a set of I/O nodes in order to facilitate parallel access. The specifics of a given file distribution are described with three metadata parameters: a base I/O node number, number of I/O nodes, and stripe size. Carns also teaches that the size specifies

that the stripe size–the unit by which the file is divided among the I/O nodes–is 64 kbytes. However, reading further, in section 3.1, Carns teaches that one issue that Carns wrestled with throughout the development of PVFS is how to present a directory hierarchy of PVFS files to application processes. Carns teaches that the method had some drawbacks, it forced system administrators to mount the NFS file system across all nodes in the cluster, which was a problem in large clusters because the limitations with NFS scaling. Second, the default teaching of NFS caused problems with certain metadata operations. These drawbacks forced Carns to re-examine their implementation strategy and eliminate dependence on NFS for metadata storage. As a result, NFS is no longer a requirement and the dependence on NFS was removed. See last paragraph of section 3.1 of Carns. Thus, Carns specifically teaches against using striping with NFS.

It should also be noted that in the background of the invention of applicants' specification, the same problem regarding PVFS is identified and specifically applicants state that PVFS filesystems are unsuitable for export over a network with NFS. See page 2, lines 1-27 of applicants' specification. Accordingly, Carns simply teaches what applicants also state in the specification, that PVFS cannot be used with NFS. The reference Carns, and its teachings, cannot be combined with the teachings of Vahalia and Fridella, because they teach away from applicants' claimed invention.

It should also be pointed out to the Examiner that applicants have more specifically defined the type of striping, with the limitation that "wherein a data file is striped among all the VFSes of the set of striped VFSes with different strips of the file's data in different VFSes in the set of striped VFSes," as found in Claim 1. This specific type of striping is not taught or suggested in the applied art of record. In addition, applicants have amended Claim 1 to further introduce structure to the striping system "wherein the striped filesystem comprises a set of striped VFSes distributed among a number of disk elements of the cluster of servers, with one VFS of the set of striped VFSes per disk element". This is not taught or suggested by the applied art of record. Moreover, applicants have specifically introduced the limitation that the striped filesystem for storing data is "for providing bandwidth to multiple disk elements", as found in claim 1. The applied art of record fails to recognize this advantage, let alone teach or suggest this limitation.

There is no teaching or suggestion in Fridella how to modify PVFS systems to overcome these problems so they are suitable for NFS. Furthermore, applicants do not suggest they discovered NFS, nor do applicants suggest they discovered striping, but applicants do take the position that they were the first to be able to apply a striped filesystem for storing data with a cluster of NFS servers. Simply identifying Vahalia and Fridella that has a plurality of NFS servers which have nothing at all to do with striping and actually have an architecture that teaches away from being used with striping, and the teachings of Carns

that specifically identify problems that make PVSF as unsuitable with NFS cannot arrive at applicants' claimed invention.

It is respectfully submitted that the Examiner is using hindsight to arrive at applicants' claimed invention. The Examiner is using the limitations of Claim 1 as a roadmap to find the different limitations in the various prior art references, and having found the limitations, concluding that applicants' claimed invention is arrived at. This is not patent law.

Accordingly, the applied art of record does not teach or suggest the limitations of Claim 1. Claims 3-13 are dependent to parent Claim 1 and are patentable for the reasons Claim 1 is patentable.

Claim 14 is patentable for the reasons Claim 1 is patentable. Claims 15-35 are dependent to parent Claim 14 and are patentable for the reasons Claim 14 is patentable.

The Examiner has allowed Claims 36 and 37.

Claim 38 has the limitation of reading the stripes of the file from each disk element having the stripes with NFS. Claim 39 has the limitation of a plurality of servers having a stripe of the VFS, and the limitation of receiving NFS write requests for a file at a

network element. For the reasons explained above regarding these limitations and Claim 1, Claims 38 and 39 are patentable over the applied art of record.

As explained above, Vahalia and Fridella have nothing at all to do with striping. It is respectfully submitted the Examiner is simply citing Vahalia and Fridella for their overall architecture, which the Examiner suggests can be used with the teachings of striping by Carns to arrive at applicants' claimed invention.

Claim 48 has the limitation "where one disk element for a given file system acts as a meta-data server". The applied art of record does not teach or suggest this limitation. Specifically, Carns simply teaches "application processes communicate directly with a PVFS manager". See section 3.1, paragraph 4. Accordingly, the use of a disk element to act as a meta data server is distinct from the teaching of Carns.

Claim 48 has the limitation that the disk element acting as a meta data server "maintains modification and change time attributes for each file". PVFS does not deal with change time attributes, or for that matter timing at all. Carns teaches that "the specifics of a given file distribution are described with three metadata parameters: base I/O node number, number of I/O nodes, and stripe size. These parameters together with an ordering of the I/O

nodes for the filesystem, allow the file distribution to be completely specified". See paragraph 2, section 3.1 of Carns. Timing is not considered by Carns.

The major problem with PVFS is its inability to handle timing issues. In fact, on page 2, lines 1-27, of the above identified patent application, applicants discuss the failure of PVFS in regard to timing issues, and how significant errors occur to the point that using a PVFS striping system is not appropriate. Carns itself admits this by teaching that PVFS is not to be used with NFS. See paragraph 5 of section 3.1 of Carns. In fact, applicants' claimed invention represents that applicants specifically discovered a way to effectively integrate timing with striping and filesystem memory applications.

Claim 48 has the limitation that "each server for each file maintains a caching element". Not only does Carns not teach or suggest this limitation, but Carns specifically teaches away from this limitation. In paragraph 5 of section 3.1 of Carns, Carns specifically teaches that the PVFS method "had some drawbacks". These drawbacks included "second, the default caching of NFS cause problems with certain metadata operations".

Claim 48 has the limitation that the teaching element "stores a last known version of the file attributes and ranges of modification time and change time values for assignment to write operation results". Again, as explained above, Carns does not deal with

times at all, let alone ranges of modification time and change time values". Most importantly, Carns specifically teaches at the end of paragraph 4, section 3.1 that "the manager is not contacted during read/write operations". In fact in this very paragraph, Carns teaches that when an application opens a file, the manager returns to the application the locations of the I/O nodes on which file data is stored. This information allows applications to communicate directly with I/O nodes when file data is accessed". In other words, this very requirement creates the uncertainty that makes PVFS unsuitable for actual use as explained by applicants on page 2 of the above-identified patent application. The fact that timing attributes are not maintained means that the wrong version of files could be accessed at given times, thus providing a user with the wrong information. This also means that the specific structure claimed by applicants, cannot be arrived at using Vahalia and Fridella since Carns specifically teaches that the application communicate directly with I/O nodes, and the architecture that the Examiner is relying upon in Vahalia and Fridella to arrive at applicants' claimed invention is in conflict with the specific teachings and requirements of PVFS by Carns.

It again must be stressed that Claim 48 has the limitation that the caching element "stores a last known version of the file attributes and ranges of modification time and change time values for assignment to write operation results". PVFS does not contact the manager during read/write operations and is not concerned with and does not teach to store the "ranges of modification time and change time values for assignment to write operation

results," as found in Claim 48. These problems with timing are one of the main reasons for the drawbacks that Carns teaches in regard to PVFS and that it cannot be used with an NFS filesystem and that the default caching caused problems, as stated in paragraph 5 of section 3.1.

Accordingly, Claim 48 is patentable over the applied art of record. Claims 49-57 are dependent to parent Claim 48 and are patentable for the reasons Claim 48 is patentable. Claim 58 is patentable for the reasons Claim 48 is patentable. Claims 59-64 are dependent to parent Claim 58 and are patentable for the reasons Claim 58 is patentable.

In view of the foregoing remarks, it is respectfully requested that the outstanding rejections and objections to this application be reconsidered and withdrawn, and Claims 1, 3-39 and 41-64, now in this application be allowed.

CERTIFICATE OF MAILING
I hereby certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on 3/5/07

Ansel M. Schwartz
Registration No. 30,587

Respectfully submitted,

Ansel M. Schwartz, Esquire
Reg. No. 30,587
201 N. Craig Street, Suite 304
Pittsburgh, PA 15213
(412) 621-9222